A Comparison of Noise Generation Techniques and the Effects on Inverse Problem Calculations

H.T. Banks
Center for Research in Scientific Computation
North Carolina State University
Raleigh, NC 27695-8205

Irene Groselj
Center for Research in Scientific Computation
North Carolina State University
Raleigh, NC 27695-8205

Abstract

To test the performance of an algorithm for an inverse problem for cases similar to those under which experimental data is used, random noise can be added to given simulated data. The most common way to do this is to use white noise generated by a uniformly or normally distributed random sequence. Another possibility is to use noise given by the so-called Rice Representation of random noise. We compare results for these two kinds of noise.

1 Introduction

Inverse or parameter estimation problems are ubiquitous in science and engineering and there exists a huge literature on computational and theoretical issues related to the development of computational methods for use in such problems (see [1] and the references therein for only a fraction of this literature). Before using associated computational algorithms with experimental data, it is common to test efficiency (or lack thereof) on simulated "data". It is well known that many algorithms work perfectly well if "exact" data (simulated "data" generated by exact solutions of the physical or biological process being investigated) is used in the algorithm (see chapter V6 in [1]). A more reasonable predictor of the algorithms' behavior when used with experimental data is that obtained when

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
1998		**00-00-1998 to 00-00-1998**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
A Comparison of Noise Generation Techniques and the Effects on Inverse Problem Calculations	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **North Carolina State University,Center for Research in Scientific Computation,Raleigh,NC,27695-8205**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**17**	

testing with "noisy" data. Hence it is desirable to add noise of some kind and to study any proposed inverse problem algorithm for robustness in the presence of noise. In this note we discuss and compare two different techniques for generation of noise to be employed in robustness investigations. The first involves the so-called Rice Representation of noise while the second, which is more widely found in the inverse problem literature, is noise produced by a random number generator. Below we describe in some detail how each type of noise is generated and then compare their use in testing an inverse algorithm in a very simple oscillator example.

2 Underlying Philosophy for the Rice Representation of Random Noise

The Rice Representation is based on the noise caused by the so-called shot effect. Shot effect noise is a typical source of noise resulting from the superposition of disturbances which occur at random.

2.1 Shot effect

If $F(t)$ is the effect at some point in the output circuit produced by the arrival of an electron at the anode of a vacuum tube at time $t = 0$, then the total effect at time t due to all electrons is given by

$$n(t) = \sum_{k=-\infty}^{\infty} F(t - t_k),$$

where the kth electron arrives at time t_k. (We assume that this series converges and the effects of electrons add linearly).

The total effect $n(t)$ can be expanded in a Fourier series (we consider only the portion of $n(t)$ lying in the interval $0 \leq t \leq T$) approximated by

$$n^N(t) = \frac{a_0}{2} + \sum_{n=1}^{N} (a_n^{N,T} \cos \frac{2\pi nt}{T} + b_n^{N,T} \sin \frac{2\pi nt}{T}),$$

where

$$a_n^{N,T} = \frac{2}{T} \int_0^T n^N(t) \cos \frac{2\pi nt}{T} dt,$$

$$b_n^{N,T} = \frac{2}{T} \int_0^T n^N(t) \sin \frac{2\pi nt}{T} dt.$$

In [4] Rice argues that the limiting Fourier coefficients $a_n^\infty, b_n^\infty, 1 \leq n \leq N$, where $a_n^\infty = \lim_{N,T\to\infty} a_n^{T,N}$, $b_n^\infty = \lim_{N,T\to\infty} b_n^{T,N}$ are independent, normally

distributed random variables. By application of the Central Limit Theorem it follows then that the noise $n(t)$, where $n(t) = \lim_{N\to\infty} n^N(t)$ is itself distributed normally.

Based on this, the original Rice Representation for random noise is given by

$$n(t) = \sum_{n=1}^{N} (a_n^\infty \cos \omega_n t + b_n^\infty \sin \omega_n t),$$

where $\omega_n = 2\pi f_n, \quad f_n = n\triangle f, \quad \triangle f = 1/T$. In this series we omit the term a_0 since we assume without loss of generality that the mean value of $n(t)$ is zero. The amplitude coefficients $a_n^\infty, b_n^\infty, 1 \leq n \leq N$ are assumed to be independent, normally distributed random variables with the following properties for the expected values (we drop the ∞ superscripts on the a_n, b_n in subsequent discussions):

- $E[a_n] = E[b_n] = 0,$
- $E[a_n b_m] = 0, \qquad$ for all n, m
- $E[a_n a_m] = E[b_n b_m] = \sigma_{f_n}^2 \triangle f \delta_{nm} = \mathcal{G}(f_n)\triangle f,$

where $\delta_{nm} = 1$ for $m = n$; $\delta_{nm} = 0$ for $n \neq m$ and $\sigma_{f_n}^2$ is the variance of a_n or b_n. $\mathcal{G}(f_n)$ denotes the power spectral density of f_n which is a measure for the power of $n(t)$ concentrated at each frequency range $\triangle f$ wide. The noise power spectrum $\mathcal{G}(f)$ is given by the sum of noise power contributions at all frequencies. $\mathcal{G}(f)$ corresponds to the distribution of the variance of the random signal $n(t)$ over frequency.

The autocorrelation function $\mathcal{R}(\tau)$ is a measure for the dependence of the random variables. One way of representing the autocorrelation function is given by

$$\mathcal{R}(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} n(t)n(t+\tau)dt.$$

From this representation we can immediately derive some distinguishing properties of the autocorrelation function:

- $\mathcal{R}(\tau)$ is an even function, i.e., $\mathcal{R}(\tau) = \mathcal{R}(-\tau)$.
- $|\mathcal{R}(\tau)| \leq \mathcal{R}(0)$, i.e., the average product is maximal if $n(t)$ is multiplied by itself.
- $\mathcal{R}(0)$ is the average value of $n^2(t)$. This value is the average power of $n(t)$ and represents the variance of $n(t)$. To get an idea of the physical meaning of "power" we have to think of $n(t)$ as a current measured in volts or amperes. Then $n^2(t)$ is the instantaneous power in watts dissipated in a 1-ohm resistor fed by $n(t)$.

The autocorrelation function and the power spectrum form a pair of Fourier transforms and are given as follows:

$$\mathcal{G}(f) = \int_{-\infty}^{\infty} \mathcal{R}(\tau) e^{i\omega\tau} d\tau,$$

$$\mathcal{R}(f) = \int_{-\infty}^{\infty} \mathcal{G}(\tau) e^{-i\omega\tau} df, \quad \omega = 2\pi f.$$

Therefore, knowledge of one of these two functions is sufficient to easily compute the missing one.

2.2 Sampling Theorems

In order to reproduce a signal without loss of information one condition concerning the spacing between the sample points has to be fulfilled. The Sampling Theorems tell us how to choose these intervals. We distinguish between the frequency and the time domain.

Sampling Theorem in the frequency domain: If a signal $x(t)$ is limited in time, i.e., $x(t)$ is nonzero only within $-T/2 \leq t \leq T/2$ and zero outside this interval, then the maximal size of the intervals between the sampling points to assure a signal without loss of information is given by $1/T$ Hz. This interval on the frequency scale is called the *"Nyquist cointerval"*.

Sampling Theorem in the time domain: If a signal $x(t)$ is restricted to a frequency band of bandwidth B Hz, i.e., the signal is nonzero only within this band and zero outside, then the maximal spacing between any two sampling points for a signal without loss of information is given by $1/2B$ sec on the time scale. This interval on the time scale is called the *"Nyquist interval"*.

2.3 Rice Representation for limited bandwidth

This representation suggested by Bendat in [2] is motivated by the original Rice Representation and is given by

$$n(t) = \sum_{n=1}^{BT} (a_n \cos \omega_n t + b_n \sin \omega_n t),$$

where $\omega_n = 2\pi f_n, \quad f_n = n\triangle f = n/T, \quad \triangle f = 1/T, \quad n = 1, 2, 3, \ldots$ and where the amplitude coefficients a_n, b_n are assumed to be independent, normally distributed random variables with the properties mentioned earlier.

For a signal being band-limited to B Hz and time-limited to T secs, we obtain (according to the Sampling Theorems) :

- In the time domain
  - The Nyquist interval is 1/2B sec long;
  - We have 2BT intervals in a record of length T.

  Therefore, we need 2BT independent numbers to completely determine the signal (one in each interval).
- In the frequency domain
  - The Nyquist cointerval is 1/T Hz wide;
  - We have BT cointervals in a band of width B.

  Again, we need 2BT independent numbers to completely determine the signal (two in each cointerval).

Since the sampling rate for the Rice Representation as suggested by Bendat fulfills the Sampling Theorem of the frequency domain, this method can be applied to generate band-limited white noise. (We have 2 random variables a_n, b_n in each cointerval $1/T$ Hz wide).

Characteristics of band-limited white noise are

- Flat power spectrum over $-B \leq f \leq B$;
- Power spectrum = 0, for $|f| > B$.

We wish to emphasize that the formal Rice Representation (with the inherent independence assumptions) can be rigorously justified only in the limiting case of white noise. As mentioned earlier the power spectrum for white noise is constant over the entire frequency spectrum. Only in this case do the Fourier coefficients a_n, b_n become completely independent. This situation, however, is physically inadmissible, since it implies infinite power of the signal (i.e., $\int_{-\infty}^{\infty} \mathcal{G}(f)df$). In real situations we are restricted to finite power. Therefore, we usually think of white noise having a flat power spectrum over the considered, limited frequency range. Only in the limiting case of white noise does the correlation function equal a Dirac Delta function indicating that the coefficients are completely independent, no matter how small are the intervals between the sample points.

3 Typical Generation of a Noise Signal Using the Rice Representation

Henceforth, we will denote noise generated as described below by RICEnoise. Using the MATLAB routine "randn(N)" which generates a normally distributed random sequence of length N with mean 0.0 and variance 1.0, we generated 2

normally distributed random sequences $a(0,\sigma^2), b(0,\sigma^2)$ for a specified σ. In MATLAB this can be done by the simple commands

$$a = \text{randn}(N) * \sigma, \quad b = \text{randn}(N) * \sigma.$$

For a specified number of time points $t, 0 \leq t \leq T$, the RICEnoise is then computed as explained in Section 2 by

$$\text{RICEnoise}(t) = \sum_{n=1}^{N} (a(n) \cos \omega_n t + b(n) \cos \omega_n t),$$

where $\omega_n = 2\pi f_n, \quad f_n = n \triangle f, \quad \triangle f = 1/T$. Here $a(n)$ and $b(n)$ correspond to a_n and b_n, respectively, in Section 2. Note that since it is not possible to generate RICEnoise with a certain variance we had to fix *one* σ for the a's and b's, which is in contrast to the original Rice Representation where the a_n's and b_n's have *different* σ_n's. It is justified to do this, since we know that the limiting Fourier coefficients in the Rice Representation are independent, normally distributed random variables. From this it follows that the RICEnoise is distributed normally (with an unknown variance, though).

For a band-limited signal generated by the Rice Representation we chose $T = 10$ secs and $B = 5$ Hz, and we sampled $N = 200$ points. In order to obtain noise of magnitude 10% through the Rice Representation we made some experiments in MATLAB. It turned out that by taking a standard deviation $\sigma = 0.0075$ for the random coefficients $a(n), b(n)$ we obtain RICEnoise of 10% magnitude. The generated RICEnoise, its power spectrum and its correlation are shown in Fig. 1 and Fig. 2, respectively. The power spectrum for the RICEnoise (see left graph in Fig. 2) is flat as expected for white noise. Fig. 2 also clearly exhibits the band-limitation. The correlation function (see right graph in Fig. 2) exhibits the typical peak at $t = 0$, where the correlation of RICEnoise(0) with itself is measured. As time increases the correlation, of course, decreases.

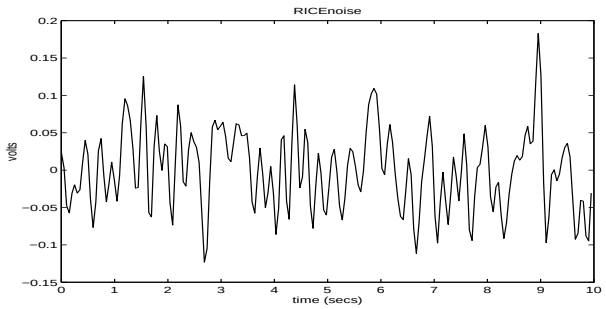

Figure 1: RICEnoise

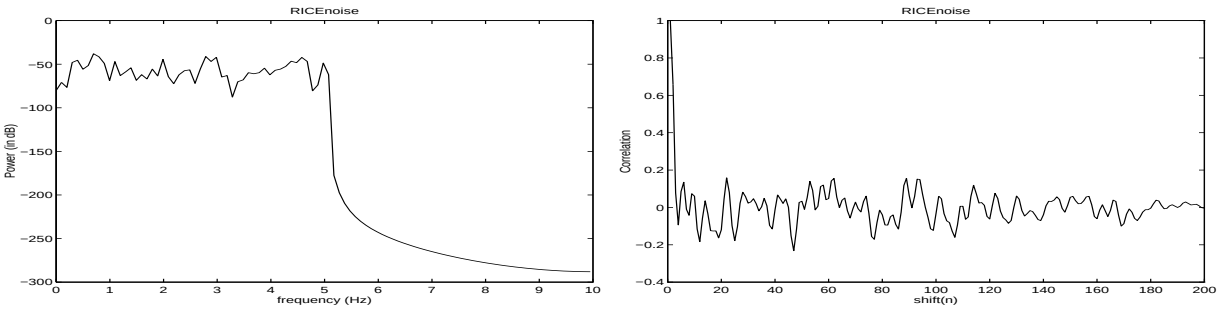

Figure 2: Power spectrum and correlation of RICEnoise

4 An Alternative Noise Generation Technique: Use of a Simple Random Sequence

In this section we shall describe generation of noise that we shall refer to as RNGnoise. Generating RNGnoise and passing it through a filter are frequently used approaches.

In order to obtain RNGnoise of magnitude 10% a random sequence with standard deviation $\sigma = 0.05$ and mean $\mu = 0$ was generated ($N = 200$ sample points). This is based on the fact that the choice of $\sigma = 0.05$ results in a normal distribution having 95% of its values in $[-2\sigma, 2\sigma]$,i.e., in $[-0.1, 0.1]$. As explained above, in MATLAB this simply can be done by using the MATLAB random number generator "randn". We obtain a sequence of N independent, normally distributed random numbers with the specified σ by

$$\text{RNGnoise} = \text{randn}(N) * \sigma.$$

The RNGnoise is shown in Fig. 3. Its corresponding power spectrum and correlation function are given in Fig. 4.

RNGnoise is white noise with a flat power spectrum over a certain frequency range. Typical plots of a white noise power spectrum and correlation function are given in Fig. 4. Note that due to the time limitation of the signal the correlation function no longer equals a Dirac function, but the samples are still statistically independent, which is different from RICEnoise. Signals with a specified finite bandwidth are usually obtained by passing white noise through an appropriate filter. The resulting noise is so-called colored noise with a non-flat power spectrum. In this case, however, the samples are no longer independent.

As we have said, the power spectrum of the RNGnoise is flat over the entire frequency range, which is typical for white noise (see left graph in Fig. 4). The correlation function of the RNGnoise (see right graph in Fig. 4) exhibits typical white noise behavior as well. We observe one peak at time t=0, where the

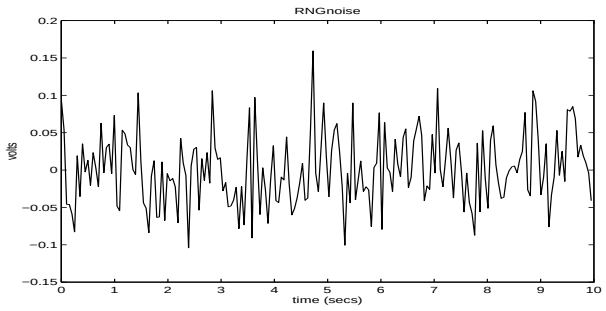

Figure 3: RNGnoise

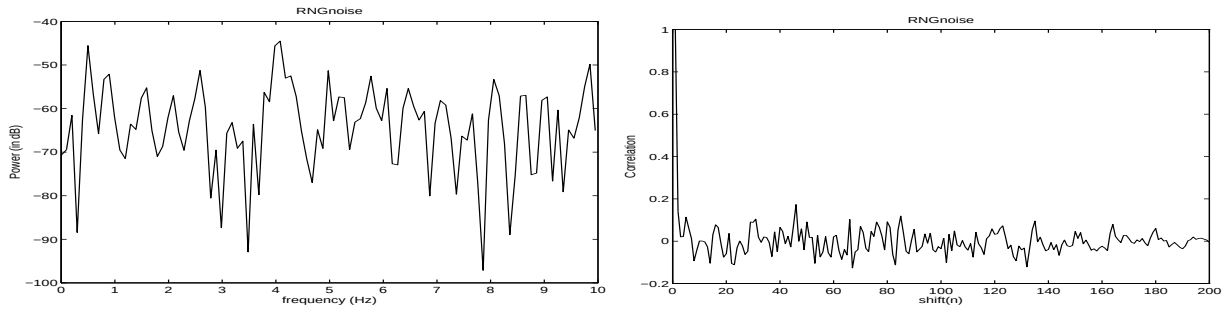

Figure 4: Power spectrum and correlation of RNGnoise

correlation of the random variable with itself is measured, which is, of course, maximal. After that the correlation decreases rapidly towards zero, since the samples generated by the random number generator are independent and normally distributed.

5 Corrupting a Signal With Noise

In the following we investigate the effect of noise on a signal. We compare the results for a signal corrupted by RICEnoise with one corrupted by RNGnoise. We distinguish between adding relative and absolute noise to a signal. The usual approach to corrupt a signal with relative noise is given by

$$\text{NsignalRel} = \text{signal} * (1+\text{noise}).$$

If the signal is to be corrupted with absolute noise, one uses

$$\text{NsignalAbs} = \text{signal} + \text{noise}.$$

Above, "NsignalRel" and "NsignalAbs" are the noisy signals. "NsignalRel" denotes the signal corrupted by relative noise, whereas "NsignalAbs" denotes the signal corrupted by absolute noise.

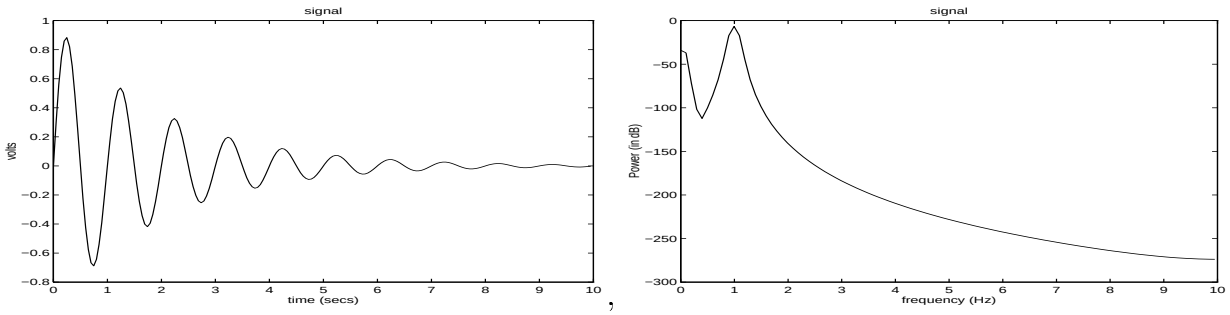

Figure 5: Signal and its power spectrum

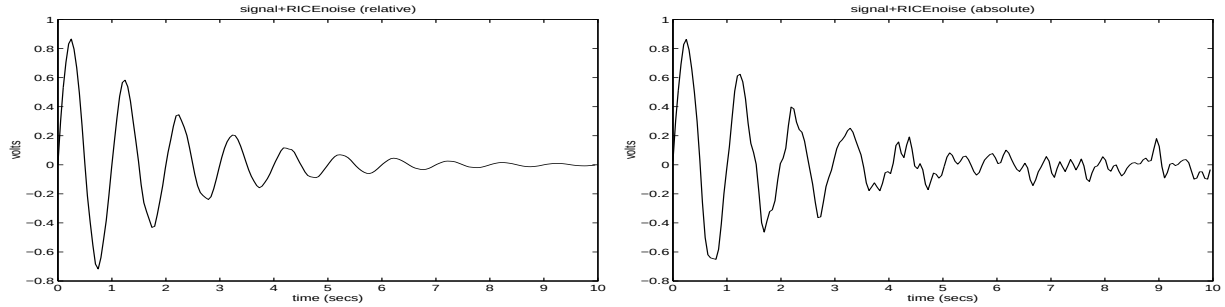

Figure 6: Signal+10% RICEnoise

The signal we used was given by

$$x(t) = e^{-0.5t} * \sin(2\pi t), \quad 0 \leq t \leq 10.$$

Plots of the signal and its power spectrum are given in Fig. 5. The disturbed signals are shown in Fig. 6 and Fig. 7. In the left graph of Fig. 7, where we added 10% relative RNGnoise, almost no difference to the undisturbed signal is visible. This situation changes if we add 10% of absolute RNGnoise, which can be seen in the right graph of Fig. 7. The disturbances are, as expected, very strong in regions where the signal is almost zero. A similar situation can be observed, if we corrupt the signal with RICEnoise (see Fig. 6).

Whereas there is no big difference visible between the signals disturbed by the RICEnoise and signals disturbed by the RNGnoise (see Fig. 6 for the RICEnoise and Fig. 7 for the RNGnoise), we can observe differences in their power spectra, as expected. This can be seen by comparing Fig. 8 for the RICEnoise to Fig. 9 for the RNGnoise.

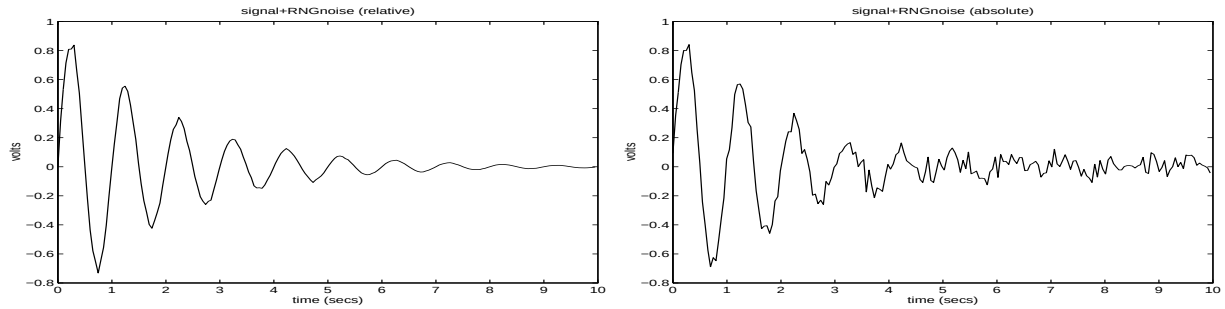

Figure 7: Signal+10% RNGnoise

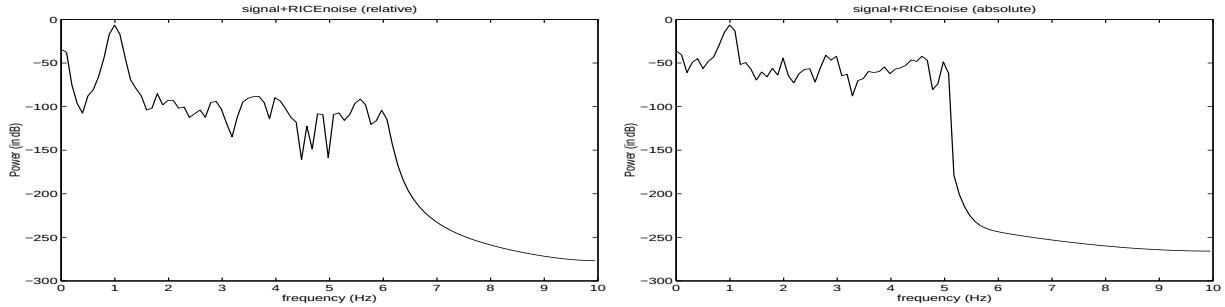

Figure 8: Signal+10% RICEnoise (power spectra)

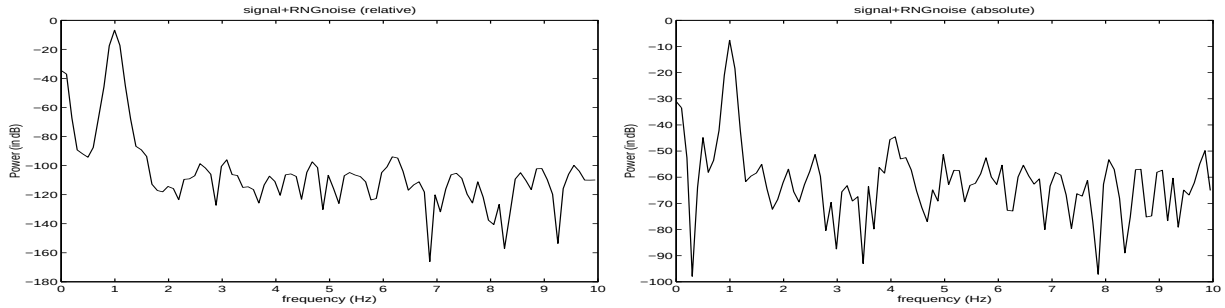

Figure 9: Signal+10% RNGnoise (power spectra)

6 Solution of an Inverse Problem

To further investigate and illustrate the differences between RNGnoise and RICEnoise we used both kinds of noise to corrupt data used to solve an inverse problem. For more information about inverse problems and parameter estimation we refer to [1]. The problem we considered is given by

$$\ddot{x} + c\dot{x} + kx = 0, \tag{1}$$

subject to initial conditions $x(0) = 1$ and $\dot{x}(0) = 0$. For values $c = 0.5$ and $k = 1.3$ the exact solution to this problem is given by

$$x(t) = e^{-0.25t}(\cos 1.1124t + 0.2247 \sin 1.1124t).$$

For the inverse problem the parameter vector is given by $q = (c, k)^T$. We chose $N = 15$ and used the exact solution to get simulated data $z(t_i), 1 \leq i \leq N, 0 \leq t_i \leq 10$ for c=0.5 and k=1.3. The inverse problem, where we want to estimate the values c and k for the parameter vector q is now stated as follows:

Find $\tilde{q} \in Q$ such that

$$J(\tilde{q}) = \min_{q \in Q} \sum_{i=1}^{N} |x(t_i; q) - z(t_i)|^2,$$

where Q is a set of admissible parameters and $x(t_i; q)$ are forward solutions of the differential equation (1) obtained by an ODE solver (we used the MATLAB routine "ode23").

In the following numerical experiments we took $q_0 = (2.6, 5)^T$ as starting value for the inverse problem. In the cases where we corrupted the given simulated data with noise to get situations similar to those where experimental data is used, the magnitude of the noise was given by 10%, i.e., we generated noise as explained in Section 3 and Section 4. The values we obtained for q after solving the inverse problem with different data are summarized in Table 1.

	Fig.	c	k
no noise added to data	10	0.4973	1.3002
data=data(1+RICEnoise)	12(left)	0.4991	1.3062
data=data+RICEnoise	12(right)	0.5108	1.3013
data=data(1+RNGnoise)	13(left)	0.5002	1.3077
data=data+RNGnoise	13(right)	0.4647	1.2780

Table 1: Results for the inverse problem

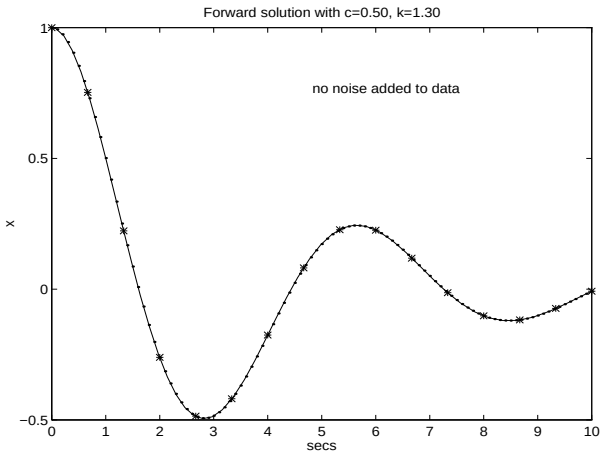

Figure 10: Exact solution

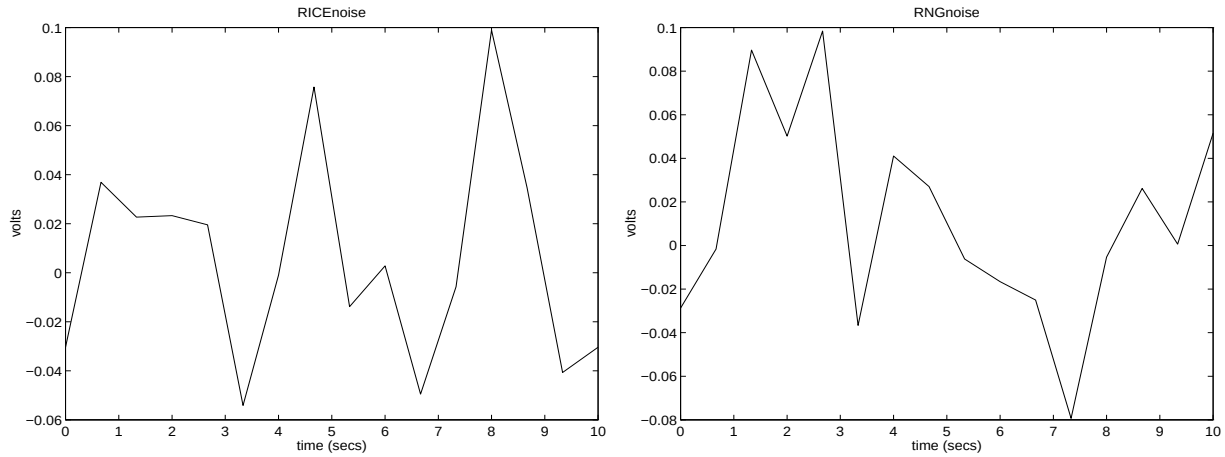

Figure 11: RICEnoise and RNGnoise

After having solved the inverse problem we used the solutions to solve the forward problem corresponding to the optimal parameter values. The results are given in the following figures. The exact solution is drawn by a dotted line, the forward solution is given by a solid line and the given data is indicated by stars.

The solution for the case where the simulated data was not corrupted with any noise, is given in Fig. 10.

Fig. 12 shows the forward solutions when the given data was corrupted with RICEnoise. The RICEnoise we used is shown in the left graph of Fig. 11. The forward solution is almost exact if relative RICEnoise is added to the simulated data (see left graph in Fig. 12). If absolute RICEnoise is added small differences between the exact solution and the forward solution are observed, which can be

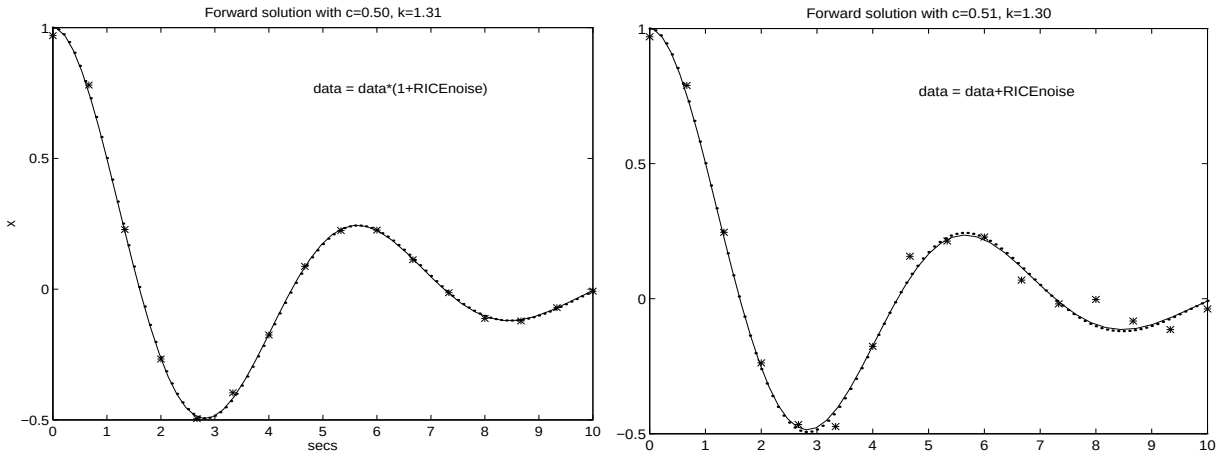

Figure 12: Results for data corrupted with 10% RICEnoise

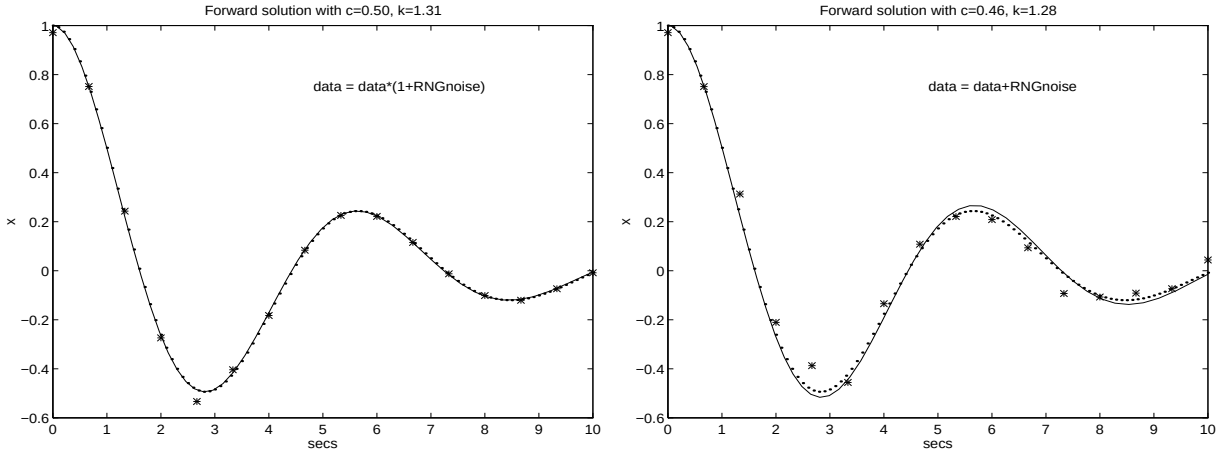

Figure 13: Results for data corrupted with 10% RNGnoise

seen in the right graph of Fig. 12.

The RNGnoise we used is shown in the right graph of Fig. 11. The results we obtained when we added this noise to the given simulated data are shown in Fig. 13. We find results similar to those we obtained for RICEnoise. The solution is almost exact if we add relative RNGnoise, but we again observe small differences between the forward solution and the exact solution when we add absolute RNGnoise to the given data.

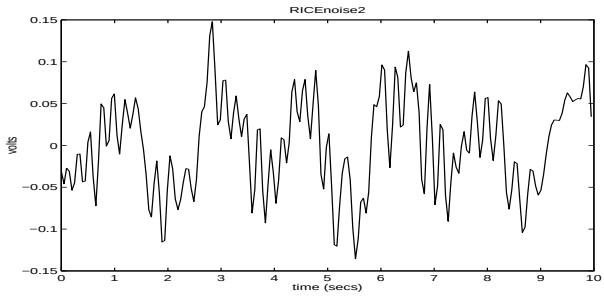

Figure 14: RICEnoise2

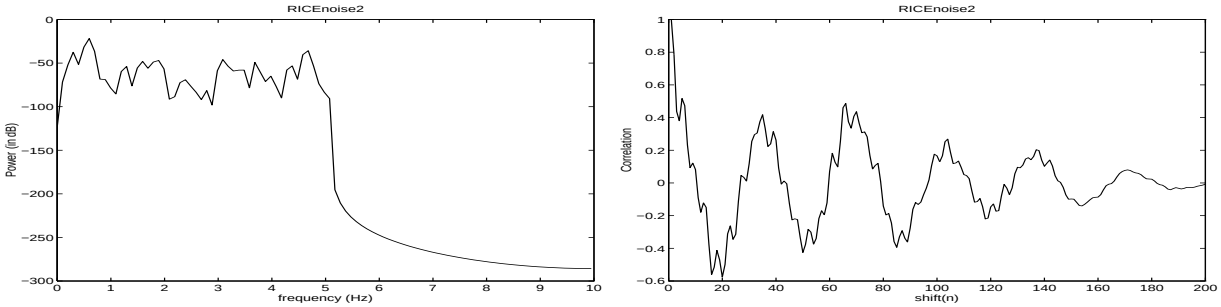

Figure 15: Power spectrum and correlation of RICEnoise2

7 Example 2: Use of a Different Package to Generate Noise According to Rice

The same investigation we performed in the previous sections was then carried out with RICEnoise generated by code provided by members in the Mathematical Products Division at Brooks Air Force Base in San Antonio. The RICEnoise generated by this code will subsequently be denoted by RICEnoise2. The main difference in the generation of RICEnoise as done in Section 3 and RICEnoise2 lies in the use of a different random number generator. In this paper we shall not give any implementation details of this package. The generated RICEnoise2 is shown in Fig. 14. The corresponding power spectrum and its correlation function are given in Fig. 15. The power spectra of RICEnoise and RICEnoise2 are very similar, whereas the correlation of RICEnoise2 is much higher than the correlation of RICEnoise (compare Fig. 15 to Fig. 2). The effects of corrupting the signal we used in Section 5 with RICEnoise2 are shown in Fig. 16 and Fig. 17. Almost no difference can be observed between RICEnoise2 and RICEnoise.

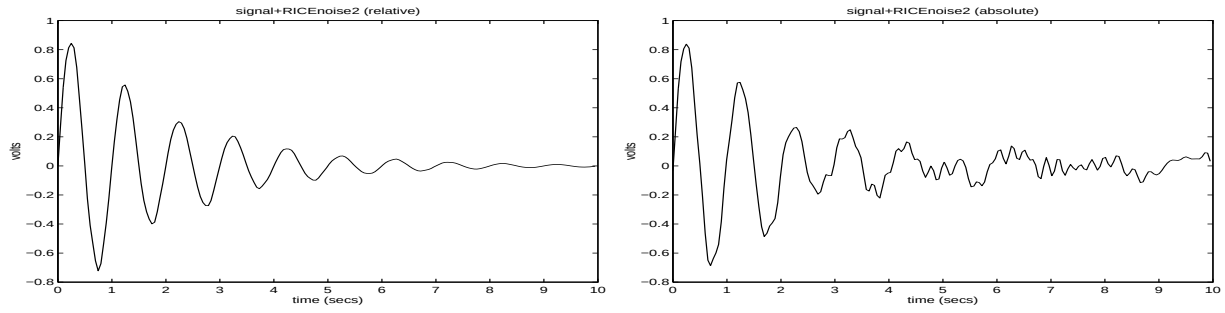

Figure 16: Signal+10% RICEnoise2

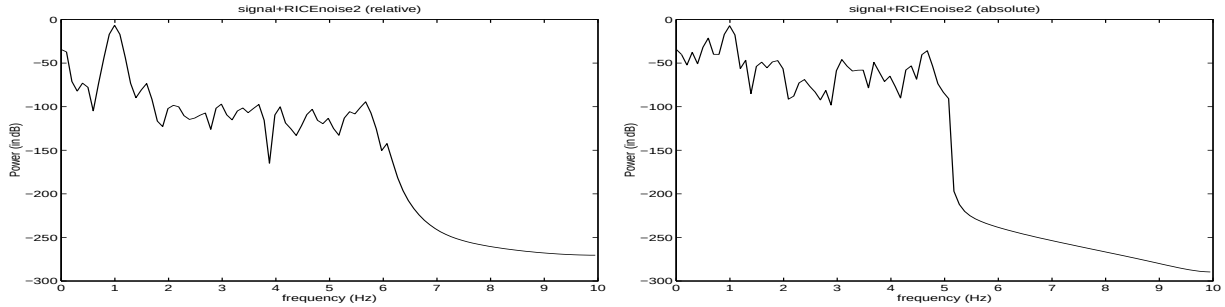

Figure 17: Signal+10% RICEnoise2 (power spectra)

We also solved the inverse problem described in Section 6 for data affected by RICEnoise2. The results of the inverse problem are given in Table 2. The parameter values summarized in Table 2 were then used to solve the forward problem. The data was corrupted with the RICEnoise2 shown in Fig. 18. As for RICEnoise and RNGnoise the solution is almost exact if relative RICEnoise2 is added to the given data, and differs a little from the exact solution if absolute RICEnoise2 is added. These results are shown in Fig. 19.

	Fig.	c	k
data=data(1+RICEnoise2)	19(left)	0.5080	1.2998
data=data+RICEnoise2	19(right)	0.4643	1.3132

Table 2: Results for the inverse problem using RICEnoise2

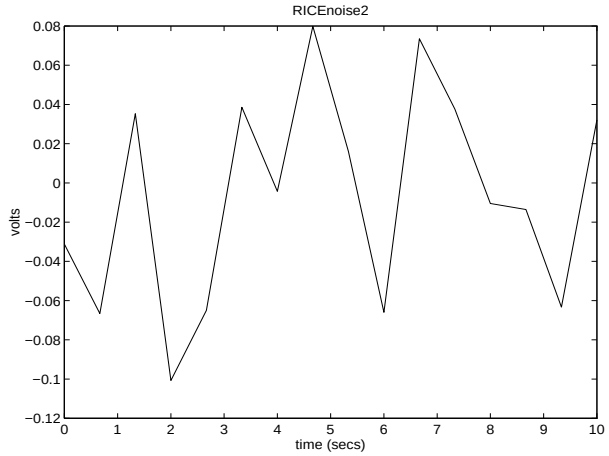

Figure 18: RICEnoise2

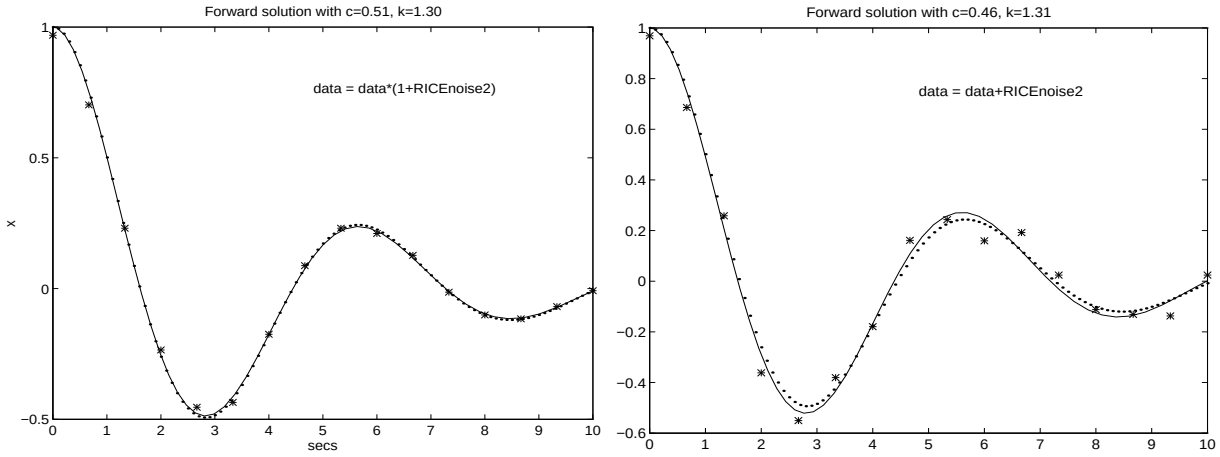

Figure 19: Results for data corrupted with 10% RICEnoise2

8 Conclusion

The Rice Representation suggested by Bendat can be used to generate band-limited white noise. We can, though, only limit the signal in the frequency space, but it is not possible to control the amplitudes of the noise signal through limited bandwidth. To obtain RICEnoise with a certain variance, one has to follow the steps described in Section 3. If we solve the inverse problem corrupting the given data with both kinds of noise we obtain similar results. We do not recognize any advantage in using RICEnoise.

Generating RICEnoise is more complicated than generating RNGnoise. The Rice Representation appears to be a very nice tool to derive some statistical properties of random noise algebraically. For practical purposes, however, especially when dealing with the time domain it becomes rather complicated. This, and the fact that the Fourier coefficients a_n, b_n are only independent in the limiting case of white noise, are possible reasons why this representation is not used very often in the engineering sciences.

9 Acknowledgment

The authors are grateful to Dr. Richard Albanese and his associates for fruitful and stimulating conversations during the course of these investigations. The research reported here was supported in part by the US Air Force Office of Scientific Research under grants AFOSR F49620-95-1-0236 AND AFOSR F49620-98-1-0180.

References

[1] Banks, H. T., Kunisch, K., Estimation Techniques for Distributed Parameter Systems, Birkhaeuser 1989.

[2] Bendat, J. S., Principles and Applications of Random Noise Theory, John Wiley & Sons, Inc. 1958.

[3] Panter, P. F., Modulation, Noise, and Spectral Analysis, Mc Graw-Hill, Inc., 1965.

[4] Rice, S. O., Mathematical Analysis of Random Noise, Bell System Tech. Journal, vol 23, pp. 282-332, July 1944; vol. 24, pp. 46-156, January 1945.

[5] Schwartz, M., Information Transmission, Modulation, and Noise, McGraw-Hill, Inc., 1970.